



TITLE:

LSI技術の展望 (ソフトウェア科学
,工学の数理的方法)

AUTHOR(S):

唐津, 修; 日比野, 靖

CITATION:

唐津, 修 ...[et al]. LSI技術の展望 (ソフトウェア科学,工学の数理的方法).
数理解析研究所講究録 1981, 436: 47-66

ISSUE DATE:

1981-09

URL:

<http://hdl.handle.net/2433/102761>

RIGHT:

LSI 技術の展望

電 電 公 社 武 蔵 野 通 研

唐 津 修

日 比 野 靖

LSI 技術の将来動向を考察するに当り、次の 4 項目を出発点に考える。

- (1) 物理的基礎科学 (マテリアル・サイエンス)
- (2) 設計技術
- (3) 製造 (プロセス) 技術
- (4) 試験診断技術

(1) 物理的基礎科学 (マテリアル・サイエンス)

マテリアル・リサーチは、現在サイエンスの領域にあり、目的意識的なエンジニアリングとしての手法が十分確立されているとは言えない。それは例えば有機化学の分野で、触媒を見付けて来る場合に一般的な法則性がなく、偶然と忍耐によって始めて発見される、といった事情と似ている。従ってマテリアル・サイエンスの 10 年後、20 年後の姿を予想する事は、他の 3 項 (設計や製

造) に比べはなはだ困難であるが、方法論的立場の見通しを立てて見るならば、1つは(ア)統計量の克服といった事であり、もう1つは(イ)極限環境の追求であると考えられる。

(ア) 統計量の克服

我々が普段接する物理現象は一般に、原子等の統計的挙動に基くマクロ的現象に他ならない。半導体中の不純物分布、金属の結晶粒径等一定の統計に従って構成されて居る。ところでこれらの統計の分散を考えると、理論モデルとして仮定した標準偏差と実際の分散との間に、大きな隔りのある場合がある。これは“製造”の技術が未熟で、理論をフォローするだけの十分な均一さで目的の物を作り上げる事が出来ないためである。そこで本当にモデルどりの均一さで物を作り上げて行くと何が起るか、又は意図的に分散をつけたらどうなるか、といった事は古くて新しい問題である。

金属の完全単結晶(ウィスカ等)の様に、放置しておけば多結晶となってしまう物を単結晶化するとか、アモルファスの様に、単結晶性を破壊してしまう事により新しい性質を導き出す、といった事も結極、統計量の克服、統計の制御という1つの思想のもとに集約される。

別の例では、スーパー・ストラクチャの生成がある。

スーパー・ストラクチャとは超格子（スーパー・ラティス）の様に通常の世界には存在しない、又は極めて存在確率の低い構造を人為的に作り出して見る事である。超格子では、数原子層毎にヘテロ・エピタキシーを積み重ねて行き、全体でマクロな超格子を実現しようとする。本来ヘテロ・エピタキシー自体、自然界には存在しにくいものであるが、その又規則的なくりかえし等まず絶無と言える。それを特別の工夫（分子線エピタキシー技術等）によって実現する事により、今迄に無い性質を持った物質を生み出そうという訳である。この事も統計の制御に他ならない。

統計を制御するとは、何やら神をも恐れぬ技の様に聞えるが、量子力学的な不確定未知の量を制御するのではなく、技術が至らないために確定未知の領域にあったものを確定既知として白日の元にさらけ出そうとする訳である。

（イ） 極限環境の追求

超高真空、超高圧、等通常的环境と著しく異なる条件のもとでの物質のふるまいを論じる事により物の性質を考えようとする。この場合極限環境下での性質は、そのまま利用する事もさる事ながら、極限環境下で発見した現象を、なるべく通常環境に近い所で実現させる様、物の

種類を変えて見たりすると更に実用価値が高まる。

物の性質は、環境パラメータに対して常にスムーズな関数形をしているとは限らないので、極限環境というのは目新しい現象を探す格好の手法である。

(2) 設計技術 (デザイン・エンジニアリング)

L S I の中に有機的に収容される素子の数を、飛躍的に増大させて行った時出現するのが、デザインの問題である。

(ア) 何故デザインか

物を作るには先ず設計 (デザイン) が必要な事は論を待たない。しかし、デザインの事を正面きって、エンジニアリングと袴まで履かせて登場させる必然があるかどうか、少し考えて見る事にする。

デザインの対象となる物が巨大化複雑化して行くと、一人のデザイナーの手には負えなくなり、多数の人間でチームを組んでデザインする事となる。この時、止むを得ずデザインを分割する事になるが、トレードオフとして必ず分割損が生じる。分割損が、単にデザイン生産性の若干の低下に留まっている内は、未だがまん出来る。ところが分割損の故に出来上った物の機能そのものが著しく低下したり、終には動かない、又はデバッグし切れな

いといった事態を招く恐れが考えられる。これでは複雑→分割の考え方は通用せず、デザイン・メソドロジーの失敗と言える。ここにエンジニアリングが必要とされる所以がある。特にLSIのデザインにおいては、論理の世界と物理（レイアウト）の世界が必ずしも1対1に対応して居らず、変な分割は命とりになる場合もある。

デザインの問題は、一方で製造工程と密接に結びついている。一度デザインを完了した物が製造後においてデバッグ可能であれば、デザインに対する完了度の要求は相当ゆるめる事が出来る。カード上でのラッピングや半田付けによる修正を思い起されれば良く分る。これに対し一度製造してしまった物はほとんど修正不可能である（現状のLSI）場合、デザインに対する完成度の要求は極めて厳しい。

即ち、デザイン・エンジニアリングの問題は、複雑大規模な機械の登場と、極めて修正困難な機械構造との掛け合せによってクローズ・アップされるに至った。

（イ） 設計技術における課題

次の3課題を考える。

A. 設計記述と言語

B. 設計データの管理

C. コンピュータ・リソースとマン・パワーの接点

A. 設計記述と言語

L S I の設計においても、設計言語を使用して作業をすすめる点は、他と同様である。大規模 L S I の設計には、構造化、階層化といった手法を活用出来る言語が開発され使用されている。しかし、L S I の設計においては各デザイン・フェーズによって記述すべき概念の内容が様々であるため、単に入れ子構造の書式のみでは不都合が起る。例えば、システム・デザインは記述出来ても電気回路は記述出来ない、とか、論理図は記述出来ても物理的レイアウトはダメという具合になりやすい。

現在いくつかの設計記述法が提案され、試みられて居るが、上に述べた意味で、一元的記述は未だ完成されていない。このため記述の変換が不都合なマッピング、不確定な部分等を生じる原因となって居り、記述体系として完結していない。しかし、問題点がある程度明確になりつつあるので、いずれ記述の統合問題は克服されるであろう。ただ、最後に残るのは一番最初の部分の正に概念を書き下す作業であり、これについてはここ当分明確な答は出ないと思われる。

B. 設計データの管理

記述言語が出来れば、設計データ自体はあまり問題に

ならない様に思われがちであるが、実際は別の次元の話である。それはあたかも国語の文法書と辞書が別の存在であるのと同じである。設計データは各々のフェーズに応じ関連を持ちながら、しかも能率良く管理されている必要がある。

大規模な装置を設計すればデータの量も当然膨大となる。これらのデータを誤り無く管理し、す早く有効に使用していく事は、もはや人手に負える仕事では無い。積極的コンピュータ・リソースの活用が不可欠である。しかし、現在最高の性能のコンピュータを使用したとしても、大規模LSIの全ての設計フェーズに関するデータを効率良く一元的にデータベース化する事は大変困難である。それはA項で述べた様に、データの属性が設計フェーズによって、多岐に渡っている事による。具体的には、①少量データのアクセスと大量データのアクセスが常に混在する事、②データの修正が頻繁である事、③データ間の相互関連が複雑である事等である。このため現在ではそのデータの種類によって何組かに分けた上、各々独立に管理する方式が多い。しかしこの事は一部のデータを修正した時、ただちに関連のある他のデータを修正するという作業を著しく困難にし、結局は誤りを混入させる原因となっている。

より能率の良いデータ管理方式や、高能力のデータ・

ベース・マシン等の開発が必要となる。また機能や負荷を適当に分散した専用システムの構築等も考慮すべきである。

C. コンピュータ・リソースとマンパワーの接点

複雑大規模化したLSIの設計において、コンピュータ・リソースの活用が不可欠である事は、多くの人が指摘している。ここで考えておくべき事は、逆に一切合切をコンピュータに任せきってしまえるか、という点である。

様々なLSIの設計フェーズの中で最後のボトルネックになるのは、配置や配線のレイアウト問題になると予想されている。その理由は第一に情報量自体が他の設計フェーズに比べて著しく多い事、第二に座標データはそれ自身記号論的な属性を持たせにくく、論理的なデバッグに極めて不便である事等の相乗効果による。そこで、コンピュータによる完全自動配置配線が有力な武器になるはずであるが、現状では未だ万能ではない。空間配置の最適化等の問題において、人間の直感とセンスをしのぐのは容易でない。しかし一方で、人間は量の増大、無謬性の確保に極めて弱い。従って、コンピュータ・グラフィック等のサポートにより人間がインタラクティブな介入をして、デザインをブラッシュアップするフェーズ

が必須となる。レイアウト以外の設計においても、コンピュータに任せきりに出来ない事情は大同小異である。

以上見てきた様に L S I 設計が内包する技術課題は、ある面では既にソフトウェアで起っている事態に類似しており、他の面では今まで我々が経験した事の無い状況を出来させている。何れにせよ、後に (4) で述べる試験診断問題と合せ L S I の大規模化を阻む大きなネックである。

(3) 製造 (プロセス) 技術

L S I の大規模化を考える時、直感的に分り易いのが製造法の話である。確かに実際の製造手法が確立されて始めて物が世に送り出されるのであるし、又微細化という事が技術の進展のバロメータの様な役割を果たして来たため、ポピュラーであったと言える。プロセス全般の推移は解説の類も多く、ここでは幾つかのポイントに触れる事としたい。

(ア) 微細化について

微細化による集積化という事が、今日の L S I の繁栄を築いている事は、疑う余地がない。将来に対する問題として、どこ迄微細化が進行し得るか、という事が最大

関心事であろう。この問題については、3つの側面から考える事が出来る。第1は、物自体をどんどん微細化した時、前と同じ機能を依持し得る限界はどこか、という事。第2は、加工技術がどこまで進歩するのか、という事。第3は、微細化に供なって減少する信号エネルギーが、熱雑音に対し十分な S/N を保てるのは、どこまでか、という事である。通常意識されている加工の限界と同時に、他の要因による限界を十分評価しておく必要がある。

一方、微細化はデバイスそれ自身のみでなく、周辺の実装部分において同様に重要である事を指摘する必要がある。例えば、現状の論理LSIは一般にピン制限を受けて居り、ピンの数が増やせないために、思った機能を集積出来ない、といった不都合が生じている。これらの問題を総合的にとらえて行かないと、将来の微細化を論じる場合、片手落となる恐れがある。

(イ) バッチ・プロセスについて

現在のLSIは、いわゆるバッチ・プロセスによって生産されている。一定のサイズと基盤材（シリコン・ウェハ等）の上に多数のチップを並べて、一斉にプロセスを通す訳である。そしてウェハ・サイズの増大にともなってバッチ当りの生産チップ数も増加している。又プロ

セス技術の改良は修熟曲線となって生産性向上をもたらす。そのおかげでチップ当り生産単価は下り、コスト・パフォーマンスが向上している。ところが一方では、バッチ・プロセスの故に歩留りという問題をかかえ、良品不良品の選別に少なからぬコスト増要因をもたらしているとも言える。

ところで、少量しか必要としない様なデバイスを作る事を考えると、いわゆるバッチ方式では都合の悪い事もある。例えば年に100個だけ生産すれば良い品種を、1バッチ1万個のラインで生産するのは、あまりに不都合であろう。LSIが大規模化して行くと、1品種当りの生産量は、減少傾向を見せる。もちろん新たな需要の開拓により、マーケットが発展して行く事を期待するにしても、絶対数が少い分野は必ず存在する。いわゆるカスタム化デバイスと称せられる物である。これらの分野のデバイスを実現する製造プロセスは、今迄のバッチ・システムとはやや趣を異にするやり方を考える事が可能である。

そもそも、バッチ・プロセスをバッチ・プロセスならしめている最大の要因は、フォト・ワークにある。フォト・リソグラフィによって多数のチップが、一回のマスク・プリントで焼きつけられるため、バッチという概念が成立している。これに対する他の手法としては、ビー

ム技術を利用した直接描き、直接デポジション、直接エッチング等であろう。制御性良く準備された各種電磁波や粒子線は、1個のチップであっても、キャンバスに絵を描く様に作り上げて行く事が出来る。

ビーム技術の長所は、ソフト・ウェアによる柔軟な制御性と、センサーとの組み合わせによりダイナミックなフィード・バックが可能な事にある。微細化、均質性の問題を乗り越えながら、しかもバッチでないプロセスを可能にする事が出来る。もちろんバッチに比べチップ当りの単価が割高になるのは止むを得ない。しかし、極端な話1個だけ必要だとしたら、1万個最少のバッチと比べて1万倍以内のコストなら許容されるはずである。又ビーム技術が駆使出来るという事は、プロセス終了したチップに再加工の道を拓くものであり、プリント板と類似の手直しを可能にさせるかも知れない。

通常考えている生産形態の他に、この様な特種プロセスの台頭がプロセス技術の大きなインパクトになる可能性を持っている。

(4) 試験診断技術

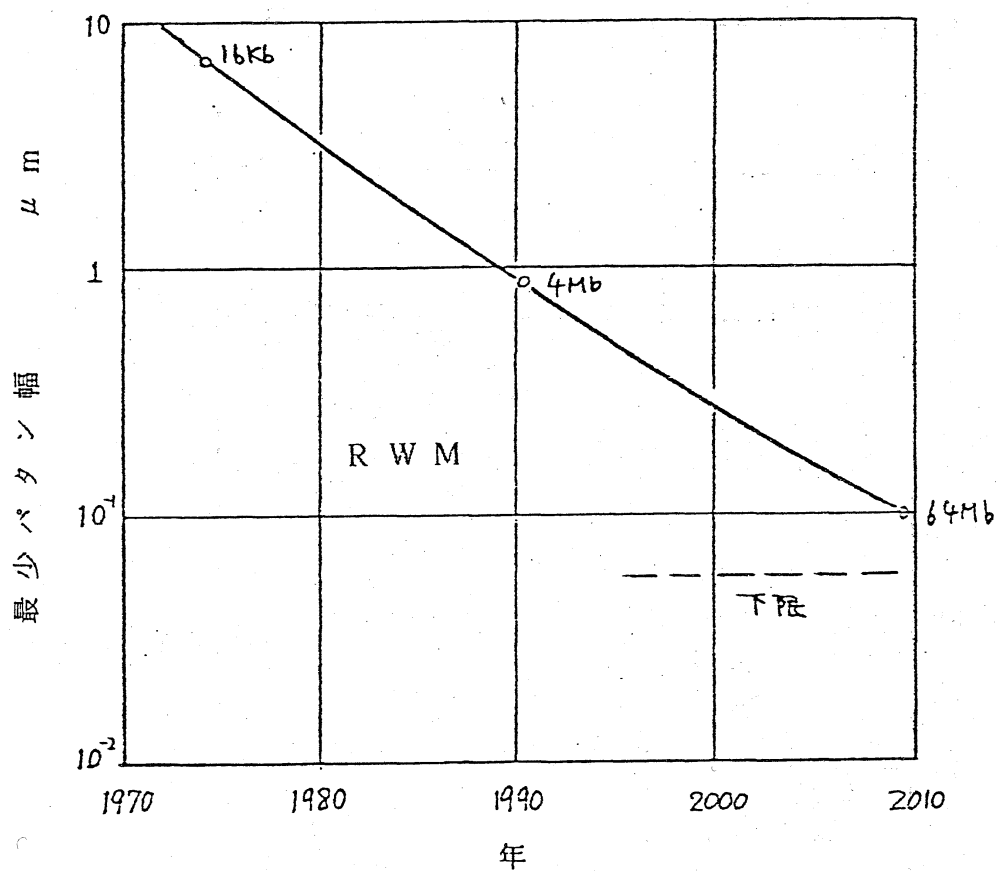
大規模なLSIにとって最終最大の問題はテストである。規模が増大した時、その機能が設計どおり正しく動作しているかどうかのテストは、膨大な作業となる。そ

してテストのためのコストが、デバイスコストの主要な部分を占めるに至る事が予想されている。又テストが不能であるが故にデザイン不能である、といった事態も十分予想される。

この問題は今迄述べてきた3項の視点いずれにも深くかかわった事であり、強いて頑張れば各項の中に埋め込んで議論する事も可能である。しかし、あえてここで別の1項として考えているのは、インテグレーション（集積）という事の本質を、テスト問題が抱えていると思われるからである。機能はデザイナーによって抽象概念を記述され、マテリアル・リサーチにより確認された現象を利用しながら、プロセスを経て生産され、ようやく現実の物として供給される。そして最初の機能スペックを本当に実現しているかどうかの確認をするのがテストである。当然、前3項全てと密接にかかわり、しかもそれらの結果の保証にもなっている。逆に言えば、テストが出来なければ、3項の努力が正当に働いて居るかどうか何とも言えない訳である。従ってあらゆる側面からテストが容易な手法を考察し、導入して行く事が必須となる。それは単にテスト容易なアーキテクチャ、回路構成の工夫に留らず、デバイス自身やプロセス等全ての部分で考慮される必要がある。

おわりに

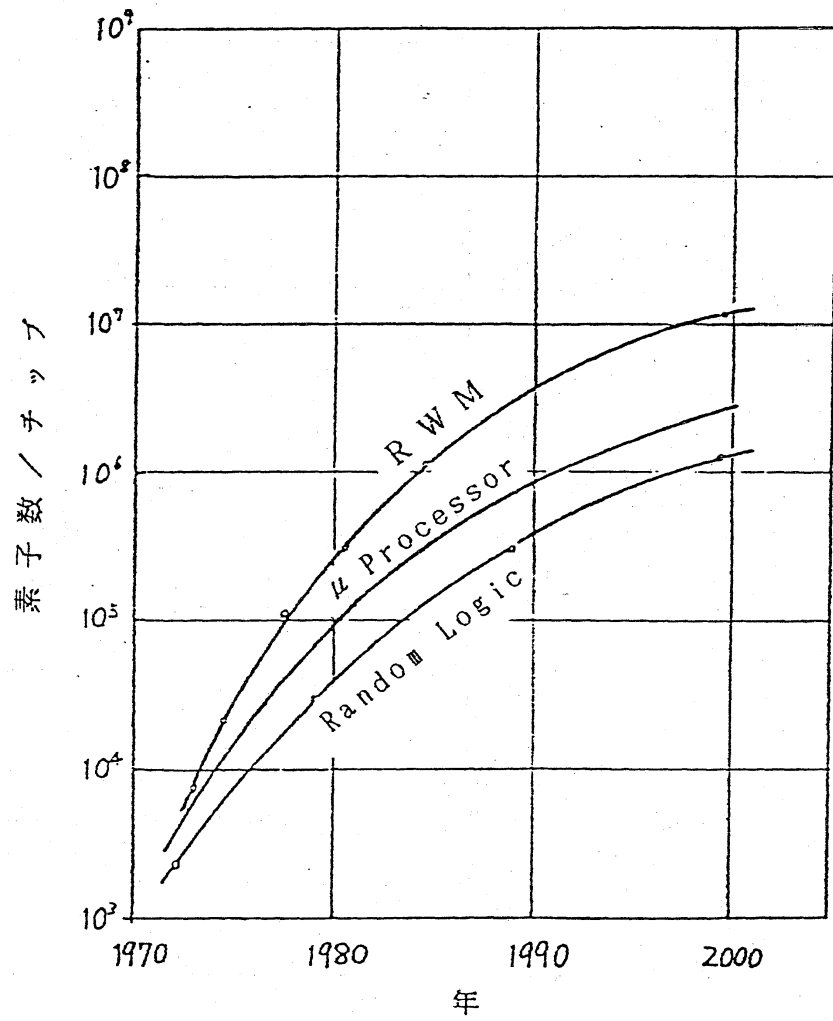
4つの項目に従ってLSIの技術課題へのアプローチ法を概観した。工業製品としてのLSIが、今後の技術開発を賄うだけの、膨大な投資を得られるかどうかは、一に掛けてマーケットの開拓努力の成否に左右される。数十万ゲートのLSIを、現在の電卓用チップ並（高々数百円）の価格で入手出来る様になった時、人類はそれを何に使うのであろうか。限り無く広がるアプリケーションの夢を、現実の形へ一つ一つ確保して行く事が、今後のLSI技術を支える重要な、そしてほとんど唯一のバック・ボーンとなるのである。



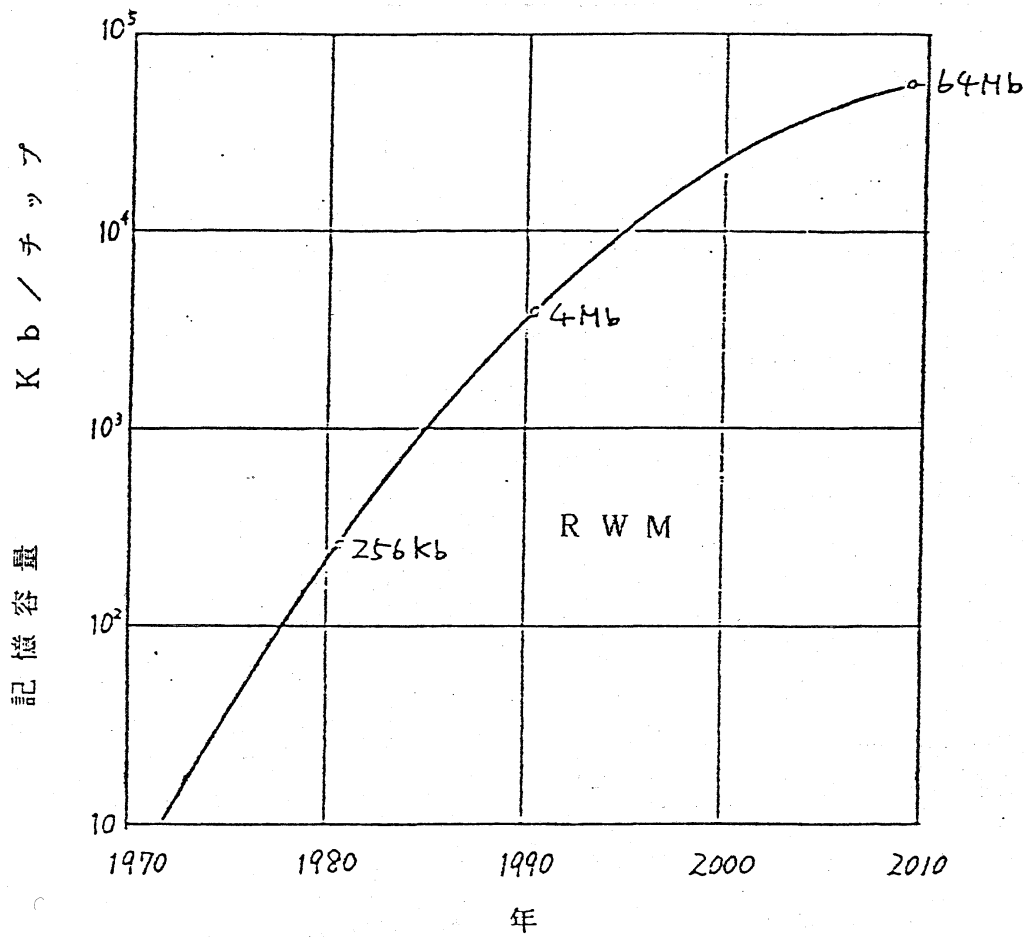
年次別 最少ボタン幅の推移

(製品導入時)

R W M : Read Write Memory



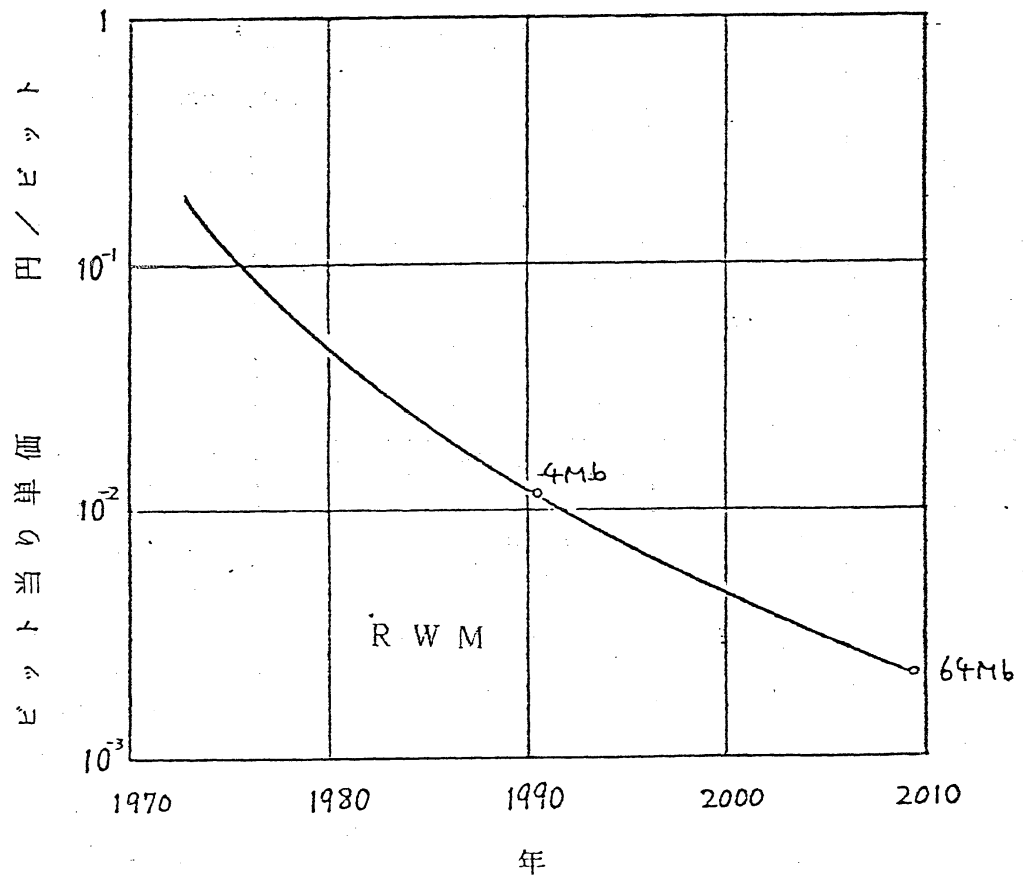
年次別 素子数 / チップの推移



年次別 記憶容量 / チップの推移

(製品導入時)

R W M : Read Write Memory

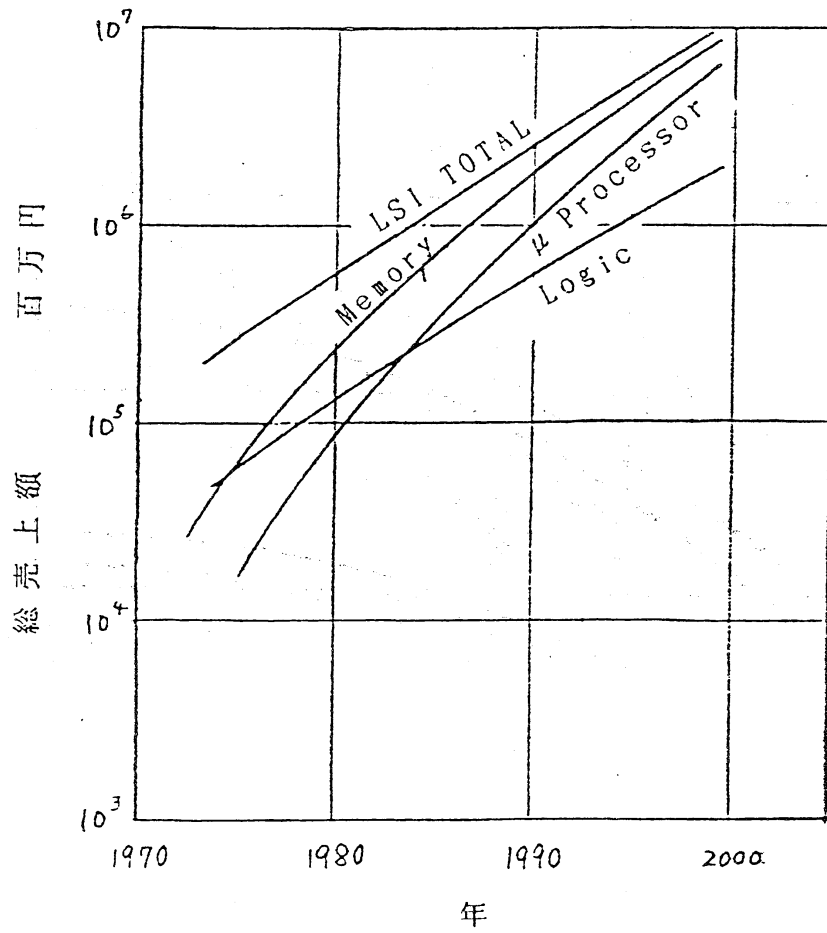


年次別 ビット当り単価の推移

(製品導入時)

注：最終試験費用は除く

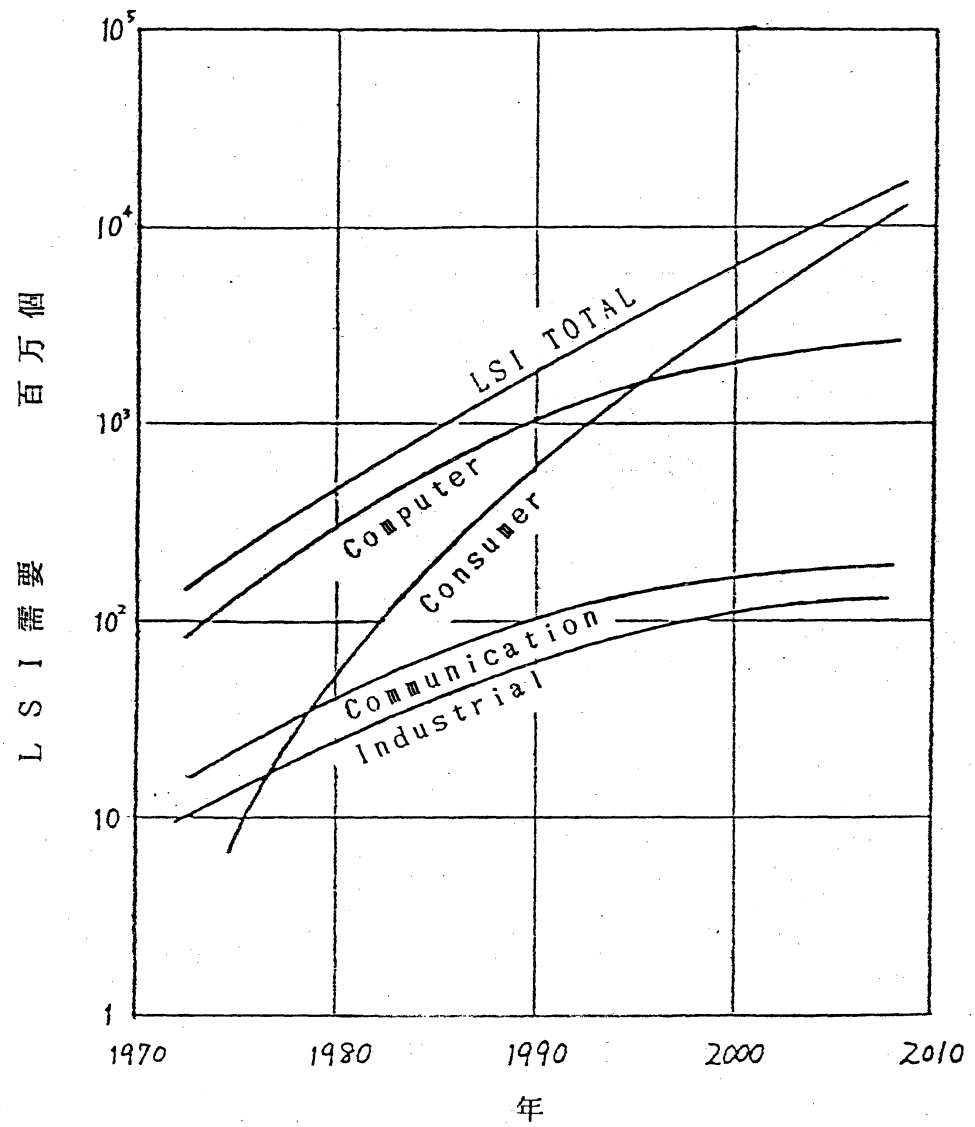
R W M : Read Write Memory



年次別 L S I 総売上額の推移

(北米と日本)

1 \$ = 200 円



年次別 L S I の使用数の推移
(北米と日本)